



Division of Informatics, University of Edinburgh

Laboratory for Foundations of Computer Science

**Stratified coherent spaces: a denotational semantics for Light Linear
Logic**

by

Patrick Baillot

Informatics Research Report EDI-INF-RR-0025

Division of Informatics
<http://www.informatics.ed.ac.uk/>

August 2000

Stratified coherent spaces: a denotational semantics for Light Linear Logic

Patrick Baillot

Informatics Research Report EDI-INF-RR-0025

DIVISION *of* INFORMATICS

Laboratory for Foundations of Computer Science

August 2000

Abstract :

Light linear logic (LLL) was introduced by Girard as a logical system capturing the class of polytime function within the proofs-as-programs approach. This paper deals with the denotational semantics of LLL: we introduce a variant of coherent spaces and prove that it is a sound model for this system, but not for usual linear logic. A simpler version of the model yields a sound semantics of Elementary linear logic, which is the analog of LLL for the class of Kalmar elementary functions.

Keywords : linear logic, computational complexity, denotational semantics, coherent spaces

Copyright © 2000 by The University of Edinburgh. All Rights Reserved

The authors and the University of Edinburgh retain the right to reproduce and publish this paper for non-commercial purposes.

Permission is granted for this report to be reproduced by others for non-commercial purposes as long as this copyright notice is reprinted in full in any reproduction. Applications to make other use of the material should be addressed in the first instance to Copyright Permissions, Division of Informatics, The University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, Scotland.

Stratified coherent spaces: a denotational semantics for Light Linear Logic

Patrick Baillot *

Laboratory for Foundations of Computer Science
University of Edinburgh, Scotland
pb@dcs.ed.ac.uk

August 28, 2000

Abstract

Light linear logic (LLL) was introduced by Girard as a logical system capturing the class of polytime function within the *proofs-as-programs* approach. This paper deals with the denotational semantics of LLL: we introduce a variant of coherent spaces and prove that it is a sound model for this system, but not for usual linear logic. A simpler version of the model yields a sound semantics of Elementary linear logic, which is the analog of LLL for the class of Kalmar elementary functions.

1 Introduction

Linear logic and time complexity. Linear logic can be seen as a typing system where one provides information about the re-usability of subprograms. This is achieved through the use of modalities called *exponentials*, distinguishing hypothesis that can be used an arbitrary number of times from hypothesis that have to be used exactly once. It follows that the exponentials typing rules constrain the information flow during execution of typed programs. Hence tame exponentials can cut down the set of typeable programs to moderate time complexity classes. Indeed in [Gir98] Girard introduced such a variant of LL called *Light Linear Logic* (LLL) which captures the class of polytime functions. A polynomial bound was given on the number

*supported by TMR “LINEAR” research network grant FMRX-CT98-0170.

of steps of the cut-elimination procedure for LLL proofs (the degree of the polynomial depending on a certain parameter of the proof called the *depth*). Conversely, any polytime function on integers was shown to be representable by an LLL proof. A simplification of this system in an affine setting (Light Affine Logic) was done by Asperti in [Asp98] and adaptations of the system to the frameworks of lambda-calculus and combinatory logic have been studied (see [Rov98] and [Abr99]).

Semantics. Still, though the motivation lying behind the restrictions on exponential rules shows quite clearly when one examines the normalisation procedure, it is not so obvious to get some *positive* intuition on these new exponentials. Light linear logic is not born with a denotational semantics as its elder sister... The supply of models could facilitate the understanding of the system, give clues to fix some points where several options are possible (e.g. auto-duality of the modality \S , provability of $!1$, adding of full weakening as in Asperti's LAL), possibly help to write representations of concrete functions.

A first step was done by Kanovich *et al* who gave in [KOS97] a semantics of *provability* through phase spaces, establishing a completeness result similar to that for LL ([Gir87]). But this approach does not provide information on the proofs themselves. An alternative idea was to look for geometry of interaction models with intrinsic complexity bound on their dynamics. Such a work was undertaken in [BP99] for Elementary Linear Logic, another variant of LL corresponding to Kalmar elementary functions (see [Gir98] and [DJ99]).

Our approach. Here we address the problem of denotational semantics for LLL. But let us first make clear what we are looking for. Indeed LLL can be encoded in LL in a natural way (compatible with cut-elimination), which is no surprise since it is a refinement of LL. Therefore any model of LL yields a model of LLL. So in fact we would like a model *specific* of LLL in the sense that it should not satisfy the principles excluded by this system (for instance the dereliction principle).

Our starting point is inspired by a semantics introduced by Martin Hofmann in [Hof99] where a size is associated to values and functions are required not to increase the size. We will instead consider sizes of *computations* defined from the number of *threads* of subcomputation opened. Morphisms will be required to offer a fixed bound on the difference between their number of output and input requests. This idea can be illustrated informally through a games semantics analogy: the AJM-like strategy ([AJM94]) associated to

the canonical proof of $!A \otimes !B \multimap !(A \otimes B)$ splits each thread of computation on the right-hand-side (output) into two threads of computation on the left-hand-side (input), so here the previously mentioned difference is equal to the size of the output and therefore is not bounded.

We will develop this idea in the context of a time-free semantics derived from coherent spaces. Actually the bounding condition will have to hold for each level of nesting of subcomputations. It then turns out that to obtain a compositional model we need to require a stronger coherence condition on morphisms, which leads us to stratified spaces and stratified cliques. Surprisingly, the stratified model (without the bounding condition) is interesting by itself as it provides a specific model of Elementary Linear Logic (ELL). It bears several similarities with the fibred phase model of [KOS97].

Recently Murawski and Ong have also proposed a games semantics for Light Affine Logic yielding a full completeness result ([MO00]). It should be interesting to try to relate their approach to the present work.

Outline of the paper. In section 2 we give preliminaries on LLL, ELL and coherent spaces, then we present the stratified coherent spaces in section 3 and show that it modelizes ELL. In 4 we define the measured spaces and the subcategory of locally bounded stratified cliques which is our model of LLL. Section 5 is devoted to the syntax of proof-nets, to their semantic interpretation and to the proof of the main soundness theorem.

Acknowledgements I am indebted to Samson Abramsky for several useful discussions along the elaboration of this work. I also wish to thank Lorenzo Tortora and Thomas Ehrhard for insightful comments and suggestions.

An extended abstract of this paper was presented at the *Second international workshop on implicit computational complexity* (ICC'00) in june 2000.

2 Preliminaries

2.1 Light linear logic

Light Linear Logic arises from restrictions on the exponential connectives of linear logic. They still enjoy the contraction $!A \vdash !A \otimes !A$ and weakening $!A \vdash 1$ principles, but:

- the *dereliction* $!A \vdash A$ and digging $!A \vdash !!A$ principles are not accepted,

- the $!$ is functorial, t.i.t.s. the rule (from $A \vdash B$ deduce $!A \vdash !B$) is accepted; it is not multifunctorial though and the principle $!A \otimes !B \vdash !(A \otimes B)$ is not allowed;
- an important point is that the equivalence between $!(A \& B)$ and $!A \otimes !B$ is maintained.

To compensate for the lack of dereliction a new modality \S is introduced, with principles $!A \vdash \S A$ and $\S A \otimes \S B \vdash \S(A \otimes B)$. Here we will not consider \S to be self-dual and we will denote its dual by $\bar{\S}$.

Girard showed how these principles could be organised into a sequent calculus ([Gir98]) which we recall below. Proof-nets were also introduced; they offer a more convenient syntax to describe the cut-elimination procedure and we will come back to them in section 5.

The *depth* of a proof-net is the maximal nesting of its exponential boxes. In [Gir98] it was shown that given a fixed depth d , a polynomial P of degree depending on d could be given such that the normalisation of any proof-net R of depth d can be performed in less than $P(|R|)$ steps (where $|R|$ measures the size of R). Conversely, the representation theorem states that any polytime function on integers can be represented by an LLL proof of the sequent $1^k; bint \vdash \S^k bint$, where:

- the formula 1^k stands for $! \dots !1$ with $!$ repeated k times,
- $bint$ is a type for integers in binary representation,
- $\S^k A$ stands for $\S \dots \S A$ with \S repeated k times.

Let us now give the sequent calculus (for the fragment without quantifiers). Light Linear Logic formulas are defined as Linear Logic formulas, but for the introduction of the new modality \S (and its dual $\bar{\S}$).

The negation is a defined one, by which we mean that A^\perp is used as a notation for the De Morgan dual of A . We also consider *discharged formulas* (denoted as $[A]$) which are expressions in waiting of contraction (they are not proper formulas and will not appear in conclusions).

A *block* is either a discharged formula $[A]$ or a multiset A_1, \dots, A_n of formulas (to be thought of as $A_1 \oplus \dots \oplus A_n$). A *sequent* is a multiset of blocks, denoted as $\vdash \mathbf{B}_1; \dots; \mathbf{B}_n$ (to be thought of as $B_1 \wp \dots \wp B_n$, where B_i is the formula associated to the block \mathbf{B}_i).

Identity group

$$\frac{}{\vdash A^\perp; A} \text{ (Axiom)}$$

$$\frac{\vdash A; \Gamma \quad \vdash A^\perp; \Delta}{\vdash \Gamma; \Delta} \text{ (Cut)}$$

Structural group

$$\frac{\vdash \Gamma}{\vdash \Gamma; [A]} \text{ (M-weakening)}$$

$$\frac{\vdash \Gamma; \mathbf{A}}{\vdash \Gamma; \mathbf{A}, B} \text{ (A-weakening)}$$

$$\frac{\vdash \Gamma; [A]; [A]}{\vdash \Gamma; [A]} \text{ (M-contraction)}$$

$$\frac{\vdash \Gamma; \mathbf{A}, B, B}{\vdash \Gamma; \mathbf{A}, B} \text{ (A-contraction)}$$

$$\frac{\vdash \Gamma; [A]}{\vdash \Gamma; ?A} \text{ (Why not)}$$

Logical group

$$\frac{}{\vdash 1} \text{ (One)}$$

$$\frac{\vdash \Gamma}{\vdash \Gamma; \perp} \text{ (False)}$$

$$\frac{\vdash A; \Gamma \quad \vdash B; \Delta}{\vdash A \otimes B; \Gamma; \Delta} \text{ (Times)}$$

$$\frac{\vdash A; B; \Gamma}{\vdash A \wp B; \Gamma} \text{ (Par)}$$

$$\frac{}{\vdash \Gamma; T} \text{ (True)}$$

$$\frac{\vdash \Gamma; A}{\vdash \Gamma; A \oplus B} \text{ (Left Plus)}$$

$$\frac{\vdash \Gamma; A \quad \vdash \Gamma; B}{\vdash \Gamma; A \& B} \text{ (With)}$$

$$\frac{\vdash \Gamma; B}{\vdash \Gamma; A \oplus B} \text{ (Right Plus)}$$

$$\frac{\vdash B_1, \dots, B_n; A}{\vdash [B_1]; \dots; [B_n]; !A} \text{ (Of course)} \quad \frac{\vdash B_1 | \dots | B_n; A_1; \dots; A_m; A}{\vdash [B_1]; \dots; [B_n]; \bar{\S} A_1; \dots; \bar{\S} A_m; \S A} \text{ (Neutral)}$$

In the (Neutral) rule, each $|$ is either a “,” or “;”.

2.2 Elementary linear logic

The system ELL is obtained by considering the same restrictions as before but allowing the principle $!A \otimes !B \vdash !(A \otimes B)$. The ELL $!$ is multifunctorial. The modality \S is not necessary in this system.

Danos and Joinet showed in [DJ99] how these requirements are fulfilled by a subsystem of plain LL proofs defined thanks to proof-net notions. They established the fact that the class of functions characterised is that of Kalmar elementary functions.

2.3 Coherent spaces

We briefly recall the main definitions on coherent spaces (see [Gir95, Gir87] for a complete exposition).

A *coherent space* is a pair $(|X|, \circ_X)$ where $|X|$ is a countable set (the *web* of X , its elements are *points*) and \circ_X is a binary reflexive and symmetric relation on $|X|$. Two elements of $|X|$ that are in this relation are said to be *coherent*. We will then write $x \circ_X y$ or $x \circ y \text{ mod } X$. We write $x \frown_X y$ if $x \circ_X y$ and $x \neq y$ (*strict coherence*). The complement of the relation \circ_X is the *strict incoherence* \smile_X and its reflexive closure is the *incoherence* relation \asymp_X .

A *clique* of X is a subset c of $|X|$ whose elements are pairwise coherent. A *multiclique* is a multiset whose underlying set is a clique.

The constructions needed to interpret linear logic formulas are the following ones:

- *negation*: $X^\perp = (|X|, \asymp_X)$,
- *tensor*: $X \otimes Y = (|X| \times |Y|, \circ_{X \otimes Y})$ where:

$$(x, y) \circ (x', y') \text{ mod } X \otimes Y \text{ if}$$

$$x \circ x' \text{ mod } X \text{ and } y \circ y' \text{ mod } Y.$$

- *with*: $X \& Y = (|X| + |Y|, \circ_{X \& Y})$ where for $x, x' \in |X|$, $y, y' \in |Y|$ we have

$$\text{inl}(x) \circ \text{inr}(y) \text{ mod } X \& Y,$$

$$\text{inl}(x) \circ \text{inl}(x') \text{ mod } X \& Y \text{ if } x \circ x' \text{ mod } X,$$

$$\text{inr}(y) \circ \text{inr}(y') \text{ mod } X \& Y \text{ if } y \circ y' \text{ mod } Y.$$

Then we set: $X \wp Y = (X^\perp \otimes Y^\perp)^\perp$, $X \oplus Y = (X^\perp \& Y^\perp)^\perp$, $X \multimap Y = X^\perp \wp Y$.

It follows that:

$$(x, y) \frown (x', y') \text{ mod } X \multimap Y \text{ iff } (x \circ_X x' \text{ implies } y \frown_Y y').$$

We are considering the multiset version of coherent spaces (it will be important in section 4): $!_m X$ is the set of finite multicliques on $|X|$ and $u \circ !_m X v$ if $u + v$ is a multiclique.

If f is a clique on $X \multimap Y$, then $!_m f : !_m X \multimap !_m Y$ is defined by:

$$\begin{aligned} !_m f &= \{([x_1, \dots, x_n], [y_1, \dots, y_n]) \text{ s.t.} \\ &\quad \forall i, (x_i, y_i) \in f \text{ and } [x_1, \dots, x_n] \in !_m X\} \end{aligned}$$

3 Stratified coherent spaces

3.1 Definitions

We introduce now a stratified version of the coherence semantics. Think of it as a way of managing *partial information* on the points of the web, t.i.t.s. on values/computations. From such a point x we want to be able to retrieve successive approximations x^0, x^1, \dots that we shall call *appearances*, providing us with increasing information about the computation. For each level of approximation a coherence relation is specified, so that the former compatibility/incompatibility single judgement on two computations is refined into a sequence of judgements, each of them corresponding to a particular level of approximation.

Definition 1 A stratified coherent space (s.c.s.) X is given by a sequence $(X^i, \phi_i)_{i \in \mathbb{N}}$ where:

- each X^i is a coherent space $(|X^i|, \subset_{X^i})$,
- each ϕ_i is a partial application from $|X^{i+1}|$ to $|X^i|$,
- the sequence is stationary: there exists an integer d such that:

$$\forall i \geq d, X^i = X^d \text{ and } \phi_i = id_{|X^d|}.$$

The least d such that the last condition is satisfied will be called the *depth* of the s.c.s. and denoted by $depth(X)$. Then X^d is called the *main space* of X and X^i is its *appearance at depth i* .

For $i \leq d$, the i -th *appearance map* is the partial application $app_i : |X^d| \rightarrow |X^i|$ given by:

$$app_i = \phi_i \circ \phi_{i+1} \circ \dots \circ \phi_{d-1}.$$

We say that an element x of X^d is *visible* if $app_0(x)$ is defined. Whenever there is no risk of confusion, given x in $|X^i|$ and $j \leq i$ we will write simply x^j for $\phi_j \circ \phi_{j-1} \circ \dots \phi_i(x)$. In particular if $i = d$ we have $x^j = app_j(x)$; we call it the *appearance of x at depth j* .

We extend app_i into a partial application from $\mathcal{P}(|X^d|)$ (resp. $\mathcal{M}(|X^d|)$) to $\mathcal{P}(|X^i|)$ (resp. $\mathcal{M}(|X^i|)$) in the natural way:

$app_i(\{x_1, \dots, x_n\})$ (resp. $app_i([x_1, \dots, x_n])$) is defined if for all $1 \leq j \leq n$ $app_i(x_j)$ is defined, in which case

$$\begin{aligned} app_i(\{x_1, \dots, x_n\}) &= \{x_1^i, \dots, x_n^i\} \\ \text{(resp. } app_i([x_1, \dots, x_n]) &= [x_1^i, \dots, x_n^i]). \end{aligned}$$

Similarly for the ϕ_i s.

We will tend to identify the stratified coherence space X with its main coherent space X^d , the reason for that being that when we consider interpretations of formulas a stratified coherent space is completely defined by its main space. In particular, $|X|$ will stand for $|X^d|$.

A convenient way to represent a s.c.s. X is as a forest \mathcal{F}_X displayed by levels:

- the trees have height $(d + 1)$, where d is the depth of X ; nodes at level $0 \leq i \leq d$ are elements of $|X^i|$;
- there is an edge between two elements a and b if one (say a) belongs to a $|X^{i+1}|$, the other belongs to $|X^i|$ and we have $\phi_i^X(a) = b$;
- the roots of the trees are the elements of $|X^0|$.

Therefore the leaves of the trees are the elements of $|X^d|$. Note that there are pending (or here rather floating) branches: their leaves are precisely the non-visible elements. With this representation, finding the i -th appearance of an element means going down the tree (from leaf to root) until reaching level i . We will see in the sequel that logical constructions include infinite branchings.

If x at level $i + 1$ is linked to y at level i we will say that x is an *immediate ascendant* of y . The *ascendance* relation is the transitive closure of this relation.

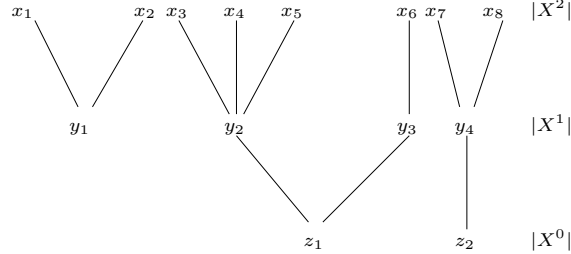


Figure 1: Example of a stratified coherent space as a forest

In the example of figure 1, x_1 and x_2 are non-visible points. The point x_3 is visible and its appearances at depth 0 and 1 are respectively $x_3^0 = z_1$ and $x_3^1 = y_2$.

Definition 2 Let X be a stratified coherent space of depth d and c be a subset of $|X^d|$. Then c is a stratified clique of X if all its points are visible and for all $i \leq d$, $app_i(c)$ is a clique of X^i .

This definition can be rephrased in our tree terminology: a stratified clique is a subforest c of \mathcal{F}_X whose trees have their root at level 0 (no floating branches!) and such that for each level i , the nodes of c at level i form a clique of X^i .

Coming back to the example of figure 1: assume $c = \{x_3, x_4, x_6, x_7\}$ is a clique of X^2 ; then c is a stratified clique of X if $\{y_2, y_3, y_4\}$ and $\{z_1, z_2\}$ are respectively cliques of X^1 and X^0 .

3.2 Constructions on stratified coherent spaces

Let X, Y be two s.c.s.. For $\square = \otimes, \&$ we define $X\square Y$ by setting for all i 's:

$$\begin{aligned} (X\square Y)^i &= X^i\square Y^i, \\ \phi_i^{X\otimes Y} &= \phi_i^X \times \phi_i^Y \text{ if } \square = \otimes, \\ \phi_i^{X\&Y} &= \phi_i^X + \phi_i^Y \text{ if } \square = \&. \end{aligned}$$

We define X^\perp by setting for all i 's:

$$(X^\perp)^i = (X^i)^\perp \text{ and } \phi_i^{X^\perp} = \phi_i^X.$$

Then $X\wp Y = (X^\perp \otimes Y^\perp)^\perp$ and $X \oplus Y = (X^\perp \& Y^\perp)^\perp$.

Finally $!X$ is given by:

- for $i \geq 0$, $(!X)^{i+1} = !_m(X^i)$, where the $!_m$ on the right-hand-side refers to the multiset bang construction on coherent spaces,
- $(!X)^0 = 1$ (the coherent space with singleton web $\{*\}$).

The applications $\phi_i^{!X} : |(!X)^{i+1}| \rightarrow |(!X)^i|$ are defined by:

- if $i \geq 1$, $\phi_i^{!X}([x_1, \dots, x_n])$ is defined if for all $1 \leq k \leq n$, $\phi_{i-1}^X(x_k) = x'_k$ is defined and if $[x'_1, \dots, x'_k]$ belongs to $|(X^{i-1})|$,
- for $i = 0$, $\phi_0^{!X}$ is the constant function equal to $*$.

Note that we have: $\text{depth}(!X) = \text{depth}(X) + 1$.

The s.c.s. $?X$ is defined as expected by: $?X = (!X^\perp)^\perp$.

These constructions on s.c.s. naturally give a notion of appearance on formulas; we only need to add that for an atomic formula α we set $\alpha^i = \alpha$ for any i .

Let us now consider an example. Let N be the stratified coherent space of depth 1 given by: $N^0 = 1$; N^1 is the set \mathbb{N} equipped with the discrete coherence relation. We denote the elements of N^1 by \underline{n} . The choice of N being of depth 1 comes from the usual representation of the integers in LL.

It follows that $!N^1 = \{k[\underline{n}], k, n \in \mathbb{N}\}$ where $k[\underline{n}]$ denotes $[\underline{n}, \dots, \underline{n}]$ with \underline{n} being repeated k times. We have:

$$k[\underline{n}] \subset l[\underline{m}] \text{ mod } !N^1 \text{ iff } n = m \text{ or } k = 0 \text{ or } l = 0.$$

Let $F = !(N \multimap N) \multimap N$. This s.c.s. has depth 3 and its appearances are given by:

$$\begin{aligned} F^3 &= !(N \multimap N) \multimap N = F \\ F^2 &= !(1 \multimap N) \multimap N \\ F^1 &= !(1 \multimap 1) \multimap N \\ F^0 &= 1 \multimap 1 \end{aligned}$$

Take x in $|F^3|$ given by $x = ([([0], 0), ([2], 0)], 1)$. It is visible and we have:

$$\begin{aligned} x^2 &= ([([*], 0), ([*], 0)], 1) \\ x^1 &= ([(*, *), (*, *)], 1) \\ x^0 &= (*, *) \end{aligned}$$

We now give an example of a non-visible point. Let $G = !N \multimap N$ and take on $|G^2|$ the points $y = ([0], 1)$ and $z = ([1], 1)$. We have:

$$y \subset z \text{ mod } G^2,$$

so $[y, z]$ belongs to $|(G)^3|$. But besides:

$$([*], 1) \smile ([*, *], 1) \text{ mod } G^1,$$

so $[y^1, z^1]$ is not a multiclique of G^1 and hence does not belong to $|(G)^2|$. Therefore, $\phi_2^{!G}$ is not defined on $[y, z]$ and so $[y, z]$ is not a visible element of $!G$.

3.3 Composition

Let f and g be stratified cliques of s.c.s. $X^\perp \wp Y$ and $Y^\perp \wp Z$. Let d be the maximum depth of these two s.c.s. and define the composition of f and g as:

$$f; g = \{(x, z) \in |(X^\perp \wp Z)^d|, \exists y \in |Y^d| \text{ s.t.}$$

$$(x, y) \in f \text{ and } (y, z) \in g\}.$$

One easily checks that $h = f; g$ is a stratified clique. Indeed, for any (x, z) of h we clearly have that $app_i(x, z)$ is defined and as $app_i(f)$ and $app_i(g)$ are cliques and $app_i(h) \subseteq app_i(f); app_i(g)$, we know that $app_i(h)$ is a clique.

This composition is naturally associative and it has identities given by: $id_X = \{(x, x), x \text{ is visible}\}$. So we have a category \mathbb{SCoH} of stratified coherent spaces and stratified cliques.

Let us stress now an important property of this composition.

Lemma 1 *If $h = f;g$ is the composition of two elements respectively of $\text{SCOH}[X, Y]$ and $\text{SCOH}[Y, Z]$, and if (a, c) belongs to $\text{app}_i(h)$ (where i is inferior to the depth of $X^\perp \wp Z$) then there exists a unique b in $|Y^i|$ such that (a, b) and (b, c) belong respectively to $\text{app}_i(f)$ and $\text{app}_i(g)$.*

Proof : The existence of b follows from the definition of the composition and from the fact that $\text{app}_i^{X^\perp \wp Y}$ (resp. $\text{app}_i^{Y^\perp \wp Z}$) is defined on all elements of f (resp. g). The uniqueness is consequence of the fact that $\text{app}_i(f)$ and $\text{app}_i(g)$ are cliques. \square

This property expresses a kind of independence of each layer: roughly speaking, the i -th appearance of the “interaction” of morphisms f and g with environment (x, z) only depends on the i -th appearances of x and z . It will be crucial when later we define a subcategory of SCOH modelizing LLL.

3.4 A model of ELL

To the previous constructions on objects correspond constructions on morphisms. For the additive and multiplicative constructions they are done in the straightforward way, layer by layer using the constructions of COH . For instance if f_1 and f_2 are respectively elements of $\text{SCOH}[X_1, Y_1]$ and $\text{SCOH}[X_2, Y_2]$ and d is the maximal depth of the s.c.s. then we have a clique $f_1 \otimes f_2$ of $((X_1 \otimes X_2) \multimap (Y_1 \otimes Y_2))^d$ and it gives a stratified clique of $(X_1 \otimes X_2) \multimap (Y_1 \otimes Y_2)$. Similarly for \wp , $\&$ and \oplus .

Let us consider now the exponential operations. Let f be an element of $\text{SCOH}[A, B]$ and let d denote the depth. We define $!f$ as:

$$!f = \{(u, v) \in !_m f, \text{ s.t. } u \text{ is visible}\}$$

where $!_m$ is the usual multiset bang functor of coherent spaces (here used in the coherent space X^d). The following lemma ensures that $!f$ is a stratified clique:

Lemma 2 *If (u, v) belongs to $!f$ then v is visible.*

It is easy to check that $!$ is a functor. We then have natural transformations given for objects A and B by co_A , w_A and $m_{A,B} \in \text{SCOH}[!A \otimes !B, !(A \otimes B)]$.

$$\begin{aligned} co_A &= \{(u + v, u, v) \in |!A^d \multimap !A^d \otimes !A^d|, \\ &\quad \text{s.t. } u + v \text{ is visible} \}, \\ w_A &= \{([], *)\}, \\ m_{A,B} &= \{([a_1, \dots, a_n], [b_1, \dots, b_n], u), \\ \text{where } &u = [(a_1, b_1), \dots, (a_n, b_n)] \in |!(A \otimes B)| \\ &\quad \text{and is visible} \}. \end{aligned}$$

If f belongs to $\text{SCoH}[A_1 \otimes \dots \otimes A_k, B]$ let us denote (abusively) by $!f$ the morphism of $\text{SCoH}[!A_1 \otimes \dots \otimes !A_k, !B]$ obtained in the obvious way from the functor $!$ and the natural transformation m .

Lemma 3 *Let $\Gamma = A_1 \otimes \dots \otimes A_k$, f belong to $\text{SCoH}[\Gamma, B]$, g belong to $\text{SCoH}[B, (B_2 \otimes \dots \otimes B_l) \multimap C]$ and let us denote by $!g$ the expected morphism in $\text{SCoH}[!B, (!B_2 \otimes \dots \otimes !B_l) \multimap !C]$, then we have:*

$$\begin{aligned} !f; co_B &= co_\Gamma; (!f \otimes !f) \\ !f; w_B &= w_\Gamma \\ !f; !g &= !(f; g) \end{aligned}$$

Proof :

These equations hold for the coherent spaces model, so here we only have to check that the constraint we added on visibility of points does not raise any problem. Let us just do it for the contraction case as an example, and assuming for simplicity that $\Gamma = A$.

Let us show first that $!f; co_B \subseteq co_A; (!f \otimes !f)$. Take an element of the left-hand side clique; it is of the form $(u, (v_1, v_2))$ where $(u, v_1 + v_2)$ belongs to $!f$. Then u is visible and there exist u_1, u_2 in $!A$ such that $u = u_1 + u_2$ and $(u_1, v_1), (u_2, v_2)$ belong to $!f$. It follows that $(u, (u_1, u_2))$ belongs to co_A and $(u, (v_1, v_2))$ belongs to $co_A; (!f \otimes !f)$.

Conversely, if $(u, (v_1, v_2))$ belongs to $co_A; (!f \otimes !f)$, then there exists (u_1, u_2) in $!A^d \otimes !A^d$ such that $(u, (u_1, u_2))$ and $((u_1, u_2), (v_1, v_2))$ are respectively elements of co_A and $(!f \otimes !f)$. Now, to deduce that $(u_1 + u_2, v_1 + v_2)$ belongs to $!f$ we need to know that $u_1 + u_2$ is visible. This is ensured by the fact that $u = u_1 + u_2$ and $(u, (u_1, u_2))$ belongs to co_A . Hence $(v_1 + v_2)$ is also visible by lemma 2, so $(v_1 + v_2, (v_1, v_2))$ belongs to co_B , $(u, (v_1, v_2))$ belongs to $!f; co_B$ and we are done. \square

Lemma 4 *There is in SCoH an isomorphism between $!(A \& B)$ and $!A \otimes !B$.*

Proof : The coherence semantics gives us a clique on the main space of the s.c.s. $!(A \& B) \multimap !A \otimes !B$ and one on the main space of $!A \otimes !B \multimap !(A \& B)$. One only has to check that these cliques, once restricted to visible elements, give stratified cliques and that they yield an isomorphism. It is the case. \square

Let us denote respectively by $is_{A,B}$ and $is'_{A,B}$ these morphisms of $\text{SCoH}[!(A \& B), !A \otimes !B]$ and $\text{SCoH}[!A \otimes !B, !(A \& B)]$.

It follows now that:

Proposition 5 *The category SCoH is a model of Elementary Linear Logic.*

The structure described on the category should already give convincing hints in favour of this statement. We shall not give a complete proof however as in section 5.4 we will prove an analogous result for Light Linear Logic, which is more interesting.

Note that the dereliction $(!A \vdash A)$ and digging $(!A \vdash !!A)$ principles are not valid in this semantics. So SCoH does not give a model of Linear Logic.

Indeed, considering the equations required and looking at the resulting cliques at depth high enough, we note that the only possible candidates would be:

$$\begin{aligned} \text{der}_A &= \{([a], a) \in |!A \multimap A| \text{ where } a \text{ is visible } \}, \\ \text{dig}_A &= \{(u_1 + \dots + u_n, [u_1, \dots, u_n]) \in |!A \multimap !!A| \\ &\quad \text{where } (u_1 + \dots + u_n) \text{ is visible } \}. \end{aligned}$$

Let us check that these are not in general stratified cliques. In fact for dig_A it is never the case if $|A|$ has at least one visible element. Indeed, given a visible element a take $x = ([a], [[a]])$ and $y = ([a, a], [[a, a]])$. Then x and y belong to dig_A and their appearances at depth 1 are: $x^1 = ([a_0], [*])$ and $y^1 = ([a_0, a_0], [*])$ in $|!A^0 \multimap !1|$.

But $([a_0], [*]) \smile ([a_0, a_0], [*]) \text{ mod } (!A^0 \multimap !1)$, and so $\text{app}_1(\text{dig}_A)$ is not a clique.

As to der_A let us assume that : $\exists a, a' \in |A|$ s.t. $a^0 \smile a'^0 \text{ mod } A^0$. Then we take $x = ([a], a)$ and $y = ([a'], a')$. These points belong to der_A and their appearances at depth 0 are: $x^0 = (*, a^0)$, $y^0 = (*, a'^0)$ in $|1 \multimap A^0|$. We have $x \smile y \text{ mod } (1 \multimap A^0)$ and so $\text{app}_0(\text{der}_A)$ is not a clique. This argument can be extended to the case where: $\exists i, \exists a, a' \in |A|$ s.t. a and a' are visible and $a^i \smile a'^i \text{ mod } A^i$. Consequently der_A is not in general a stratified clique.

3.5 Example of refutation: the iterator

As an example, we will show that the iteration principle is excluded in SCoH . Indeed it is not provable in ELL, which is normal since the class of elementary functions is not closed under iteration. Actually, strictly speaking we are not *refuting* the iteration principle as we assume a particular interpretation of integers, choose the candidate clique to interpret the iteration and show that it does not satisfy our stratification condition. But as this candidate clique is the natural one, we consider that this is already a good test for our model.

To interpret the (tally) integers we choose the stratified coherent space N of depth 1 given by: $N^0 = 1$; N^1 is the set \mathbb{N} equipped with the discrete coherence relation.

The clique $Iter_A$ would be defined on:

$$F = A \otimes !(A \multimap A) \otimes N \multimap A,$$

by:

$$Iter_A = \{(a_0, [(a_0, a_1), \dots, (a_{n-1}, a_n)], n, a_n) \in |F|\}.$$

Let us take the example of $A = N$. Take two elements of $Iter_A$ obtained with the same integer n :

$$\begin{aligned} x &= (a_0, [(a_0, a_1), \dots, (a_{n-1}, a_n)], n, a_n) \\ x' &= (a'_0, [(a'_0, a'_1), \dots, (a'_{n-1}, a'_n)], n, a'_n) \end{aligned}$$

Say we chose $a'_0 = a_0$, but $[(a_0, a_1), \dots, (a_{n-1}, a_n)]$ and $[(a'_0, a'_1), \dots, (a'_{n-1}, a'_n)]$ (i.e. the iterated functions) such that $a_n \neq a'_n$. For a concrete example one can consider $x = (0, [(0, 0)], 1, 0)$ and $x' = (0, [(0, 1)], 1, 1)$. Then at depth 1 we have:

$$\begin{aligned} x^1 &= (a_0, n[*], n, a_n) \\ x'^1 &= (a_0, n[*], n, a'_n) \end{aligned}$$

(where $n[*]$ stands for $[\dots, *]$ where $*$ is repeated n times). Then as $a_n \sim a'_n \text{ mod } N$, we have: $x^1 \sim x'^1 \text{ mod } F^1$. Hence $Iter_A$ is not a stratified clique.

The following modified iteration principle is valid in SCoH though and is provable in ELL :

$$\begin{aligned} ItEL_A &: !A \otimes !(A \multimap A) \otimes N \multimap !A \\ ItEL_A &= \{(\sum_{j=1}^m [a_0^j], \sum_{j=1}^m u_j, n, \sum_{j=1}^m [a_n^j]) \in |F|, \\ &\quad \text{where } m \in \mathbb{N} \text{ and for } 1 \leq m, \\ &\quad u_j = [(a_0^j, a_1^j), \dots, (a_{n-1}^j, a_n^j)]\}. \end{aligned}$$

4 Locally bounded stratified cliques

4.1 Measured coherent spaces

We want to enrich our semantical structure with a quantitative feature, a measuring function. As said in the introduction our goal here is to keep track of the I/O balance within a computation of the program. We mean by I/O balance the difference between the number of times the program has been called and the number of times it has requested an input. This balance will be evaluated at each level.

A *measured* coherent space X is a coherent space given together with a measuring function: $s^X : |X| \rightarrow \mathbb{Z}$. We will consider *stratified measured coherent spaces* (s.m.c.s.) (X^i, s_i, ϕ_i) adapted from definition 1. Actually spaces will be measured only starting from level 1 and s_i will denote the measure of X^{i+1} : $s_i : |X^{i+1}| \rightarrow \mathbb{Z}$ (one can think of X^0 as having constant measure s_{-1} equal to 0). The third condition is strengthened to:

there exists an integer d such that:

$$\forall i \geq d, X^i = X^d, s_{i-1} = s_{d-1} \text{ and } \phi_i = id_{|X^d|}.$$

The reason for this mismatch on indexes is that what we really care about is the size of the multisets corresponding to exponentials, and in the interpretation of formulas these only appear from level 1. We will use upper-scripts to indicate to which s.m.c.s. the measure function is relative: s_i^X is defined on $|X^{i+1}|$ where X^{i+1} is the level $i+1$ of X . We introduce the following measure functions for the various constructions:

$$\begin{aligned} s_i^{X^\perp} &= -s_i^X, \\ s_i^{X \square Y}(x, y) &= s_i^X(x) + s_i^Y(y) \text{ for } \square = \otimes, \wp, \\ s_i^{X \square Y} &= [s_i^X, s_i^Y] \text{ for } \square = \&, \oplus, \\ s_i^{\mathbf{1}}(*) &= s_i^\perp(*) = 0 \end{aligned}$$

The main case is that of the $!$ construction, for which we set: $s_0^{!X}([x_1, \dots, x_n]) = n$ and $s_{i+1}^{!X}([x_1, \dots, x_n]) = \sum_{k=1}^n s_i^X(x_k)$.

Note that by duality we then have

$$s_0^{?X}([x_1, \dots, x_n]) = -n$$

and

$$s_{i+1}^{?X}([x_1, \dots, x_n]) = \sum_{k=1}^n s_i^X(x_k).$$

As to the new modality \S we define the s.m.c.s. $\S X$ by:

- for $i \geq 0$, $(\S X)^{i+1} = X^i$,
- $(\S X)^0 = 1$.

Actually, it will be more convenient to denote elements of $|(\S X)^{i+1}|$ as singleton elements over $|X^i|$: $|(\S X)^{i+1}| = \{[x], x \in |X^i|\}$.

The measure functions are defined by: $s_0^{\S X}([x]) = 1$, and $s_{i+1}^{\S X}([x]) = s_i^X(x)$.

Note that \S is not self-dual, contrarily to the definition in [Gir98]. We shall denote its dual by $\bar{\S}$. A possibility to have a self-dual \S would be to set $s_0^{\S X}$ equal to the constant zero function.

4.2 Locally bounded stratified cliques

Definition 3 Let f be a stratified clique on a m.s.c.s. X . We say that f is locally bounded if for any x of f , for any i the following integers set is bounded:

$$\{s_i(y^{i+1}), y \in f \text{ s.t. } y^i = x^i\}.$$

Recall the forest definition of a stratified clique c given in section 3.1. Now, this forest c is a locally bounded stratified clique if to *each node* (at level i) we can associate an integer M such that the size s_i of its immediate ascendants is bounded by M (in absolute value).

Notice that if all branchings of the forest c are finite, then the condition is trivially satisfied and so c is a locally bounded stratified clique. But this will not often be the case for cliques interpreting proofs (in the case of a cut-free proof it would mean that it does not make any use of !-promotion).

Another particular case of locally bounded clique is that where the measures are always zero: $\forall x \in f, \forall i \in \mathbb{N}, s_i(y) = 0$. This condition is satisfied for instance by the identity maps, so they are bounded. It seems to be a fairly degenerate case, but surprisingly enough it turns out that it would suffice to account for all constructions of LLL ... but the §. Yet, as the § is essential to get a significative expressivity we definitely cannot stick to this particular case.

Lemma 6 *Identities, contraction and weakening morphisms and the morphisms $is_{A,B}$ and $is'_{A,B}$ are locally bounded cliques.*

Proof : One can check that they satisfy the property we just pointed out: the size of all their elements at any depth is always zero. \square

Proposition 7 *Locally bounded stratified cliques are preserved by composition.*

Proof : Let f and g belong respectively to $\text{SCOH}[X, Y]$ and $\text{SCOH}[Y, Z]$ and assume they are locally bounded. Let i be an integer, (x, z) be an element of $f;g$ and let us show that $f;g$ satisfies the property w.r.t. (x, z) at depth i .

By definition of the composition, there exists a y in $|Y|$ such that (x, y) and (y, z) belong respectively to f and g . Let us denote:

$$\begin{aligned} M_1 &= \sup\{|s_i(x', y')|, (x', y') \in f \text{ s.t.} \\ &\quad \text{app}_i(x', y') = \text{app}_i(x, y)\}, \\ M_2 &= \sup\{|s_i(y', z')|, (y', z') \in g \text{ s.t.} \\ &\quad \text{app}_i(y', z') = \text{app}_i(y, z)\}. \end{aligned}$$

We claim that $(M_1 + M_2)$ provides a suitable bound for $(f; g)$ and (x, z) at depth i . Indeed, let (x', z') be an element of $f; g$ such that $app_i(x', z') = app_i(x, z)$. There exists a y' such that (x', y') and (y', z') belong respectively to f and g . As $app_i(x', z') = app_i(x, z)$ it follows from lemma 1 that $app_i(y) = app_i(y')$ and therefore we have:

$$app_i(x', y') = app_i(x, y) \text{ and } app_i(y', z') = app_i(y, z).$$

Hence $|s_i(x', y')| \leq M_1$, $|s_i(y', z')| \leq M_2$, and as:

$$s_i(x', z') = s_i(x', y') + s_i(y', z'),$$

we conclude that: $|s_i(x', z')| \leq (M_1 + M_2)$. \square

Note that lemma 1 is used in a crucial way in the proof. If we were to define a stratified relational model in the lines of the stratified coherence model and then a notion of locally bounded stratified relations, these morphisms would not be preserved by composition.

4.3 The model of LLL

We denote by \mathbb{BSCOH} the category of s.m.c.s. and locally bounded stratified cliques. We will consider it as a subcategory of \mathbb{SCOH} even if in fact the objects are not the same (we added the measuring functions). Let us illustrate now how the quantitative condition rules out monoidality of $!$.

Lemma 8 *The stratified clique $m_{A,B}$ of $\mathbb{SCOH}[!A \otimes !B, !(A \otimes B)]$ is not locally bounded.*

Indeed for any n we have in $m_{A,B}$ an element of the shape:

$$x = (([a_1, \dots, a_n], [b_1, \dots, b_n]), [(a_1, b_1), \dots, (a_n, b_n)])$$

(take for instance the a_i 's equal to a single element a of $|A|$, and similarly for the b_i 's).

We have: $s_0(x) = (-n) + (-n) + n = -n$. So the set $\{s_0(y), y \in m_{A,B}\}$ is not bounded and therefore $m_{A,B}$ is not a locally bounded stratified clique. Consequently, the category \mathbb{BSCOH} is not a model of ELL.

Proposition 9 *The subcategory \mathbb{BSCOH} is stable by \otimes , \wp , \oplus and $\&$.*

For instance if f, f' belong respectively to $\mathbb{BSCOH}[X, Y]$ and $\mathbb{BSCOH}[X', Y']$ it is easy to check that the stratified clique $f \otimes f'$ of $\mathbb{SCOH}[X \otimes X', Y \otimes Y']$ is also locally bounded.

Proposition 10 *The subcategory \mathbb{BSCOH} is stable by the functor $!$.*

In this proof we once again use the coherence relations.

Proof : Assume f belongs to $\mathbb{BSCOH}[X, Y]$ and let us show that then $!f : !X \multimap !Y$ is locally bounded. Take (u, v) an element of $!f$. Then u and v are of the form $u = [x_1, \dots, x_n]$, $v = [y_1, \dots, y_n]$, where for $1 \leq k \leq n$, (x_k, y_k) are elements of f . Let us set a depth i . As f is locally bounded we know that for each k the set:

$$\{|s_i(x', y')|, (x', y') \in f \text{ and } app_i(x', y') = app_i(x_k, y_k)\}$$

is bounded. Denote by M_k its upper bound. Now we claim that $\sum_{i=1}^n M_k$ provides a suitable bound at depth $(i+1)$ for $!f$ and (u, v) . Indeed, take (u', v') in $!f$ such that $app_{i+1}(u', v') = app_{i+1}(u, v)$. Then it means that u' and v' are of the form $u' = [x'_1, \dots, x'_n]$, $v = [y'_1, \dots, y'_n]$ with:

$$\forall 1 \leq k \leq n, app_i(x_k) = app_i(x'_k), app_i(y_k) = app_i(y'_k).$$

Besides, as (u', v') belongs to $!f$ there is a permutation $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ such that:

$$\forall 1 \leq k \leq n, (x'_{\sigma(k)}, y'_k) \in f.$$

Note that here we cannot assume that σ is the identity as the indexing of the multiset elements has already been chosen so that elements of same indexes in u and u' (resp. v and v') have same appearance at depth i .

Let us fix one k . As $app_i(f)$ is a clique and contains (x_k^i, y_k^i) and $(x_{\sigma(k)}^i, y_k^i) = (x_{\sigma(k)}^i, y_k^i)$ we have:

$$(x_k^i, y_k^i) \odot (x_{\sigma(k)}^i, y_k^i) \text{ mod } (X^i \multimap Y^i),$$

Now, as u is visible we know that $[x_1^i, \dots, x_n^i]$ is a multiclique of X^i , and so in particular:

$$x_k^i \odot x_{\sigma(k)}^i \text{ mod } X^i.$$

Therefore $x_k^i = x_{\sigma(k)}^i$. So finally $(x_k^i, y_k^i) = (x_k^i, y_k^i)$ and consequently the bound applies:

$$|s_i(x'_k, y'_k)| \leq M_k.$$

Hence we have:

$$|s_{i+1}(u', v')| = \left| \sum_{k=1}^n s_i(x'_k, y'_k) \right| \leq \sum_{k=1}^n M_k.$$

As to the depth 0 we clearly have a bound equal to zero. \square

Let us define the action of \S on morphisms. Given f in $\text{SCoH}[X, Y]$ we set:

$$\S f = \{([x], [y]) \in |\S X \multimap \S Y|, (x, y) \in f\}.$$

It is clear that if f is locally bounded, then $\S f$ is locally bounded. This gives us a functor on BSCoH and we have natural transformations mp and nd defined by:

$$\begin{aligned} mp_{A,B} &= \{([a], [b]), [(a, b)]) \in \\ &\quad |\S A \otimes \S B \multimap \S(A \otimes B)|\}, \\ nd_A &= \{([a], [a]) \in |\S A \multimap !A|\}. \end{aligned}$$

Theorem 11 (Soundness) *The category BSCoH is a model of Light Linear Logic.*

Now, to prove this theorem we need first ... to agree on a syntax for proofs with a suitable normalisation process. Unfortunately the sequent calculus recalled in 2.1 is not satisfactory for this purpose; cut-elimination in it is just too complicated to handle. That is why we will establish the result for LLL proof-nets, for which Girard proved the polytime complexity bound in [Gir98]. Hence we need to postpone the proof to section 5.4, after we have given the definition of proof-nets and their semantic interpretation via *experiments*.

However it is worth noting already that certain extra principles are validated in this semantics, like $\S(A \otimes B) \multimap \S A \otimes \S B$. This principle was also considered by Hofmann and Jagadeesan in recent work.

4.4 Examples

The ELL iterator. We want to show that the ELL iterator $ItEL$ recalled in section 3.5 is excluded in BSCoH . We need first to set a measure for the space N . Once again guided by the LLL representation of integers we set:

$$\forall n \in |N|, s_0(\underline{n}) = 1 - n.$$

To simplify we consider elements of $ItEL_A$ of the form:

$$x = (m[a_0], mu, \underline{n}, m[a_n]),$$

(this is the particular case when all the a_0^j 's are equal and so are the u_j 's). We then have:

$$\begin{aligned} x^0 &= (*, *, *, *), \\ x^1 &= (m[*], mn[*], \underline{n}, m[*]). \end{aligned}$$

Hence:

$$\begin{aligned} s_0(x^1) &= -m - mn - (1 - n) + m \\ &= n(1 - m) - 1 \end{aligned}$$

So we might fix for instance $m = 2$ and choose elements x with arbitrary large n 's which shows that *ItEL* is not a locally bounded stratified clique.

Notice that if m is constrained to be 1, this argument does not apply. This is what happens for the LLL iterator, as the right-hand side $!$ is replaced by a \S .

The exponential function. The exponential function $n \rightarrow 2^n$ is representable in ELL. Recall that the type of tally integers in ELL is $N_X = !(X \multimap X) \multimap !(X \multimap X)$. Here we are not considering second order quantifiers, so our integer type needs to be instantiated on a particular formula X . Then the exponential function is computed by an ELL proof corresponding to the lambda-term:

$$exp = \lambda x(x) \lambda f \lambda y(f)(f)y.$$

The type of the proof is $N_{A \multimap A} \multimap !N_A$. Danos and Joinet showed in [DJ99] how the syntax of LL proof-nets can be used to denote ELL proofs. This is the representation we use to give the proof on figure 2, using as abbreviation $B = !(A \multimap A)$. Note that this syntax should not be confused with that of the LLL proof-nets ([Gir98]) that we will recall in section 5.

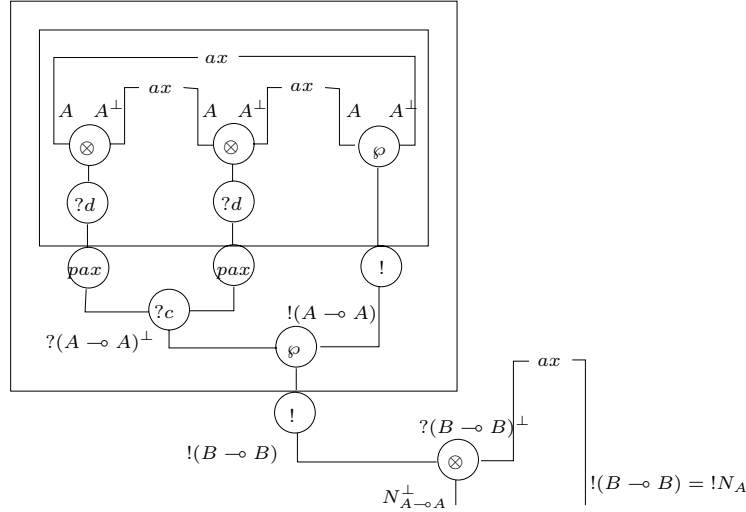


Figure 2: Proof-net R for the term *exp*.

The interpretation R^* is a stratified clique which is not locally bounded. Indeed, its elements are of the form:

$$x = (([\alpha_1, \dots, \alpha_n], \beta), \beta) \in |(! (B \multimap B) \multimap ! (B \multimap B)) \multimap ! (B \multimap B)|,$$

where $[\alpha_1, \dots, \alpha_n]$ belongs to the interpretation of the external box. We have $s_0(x) = n$, and as R^* has elements with arbitrary large n 's the condition is not satisfied at level 0.

Note however that we could remedy to this problem by typing the algorithm as $(!1 \otimes N_{A \multimap A}) \multimap !N_A$. This amounts to add to the proof-net a \perp node (followed by dereliction and promotion) at depth 1 with an adequate jump. Call R' this new proof-net. Now the element of R'^* are of the form:

$$x = (n[*], ([\alpha_1, \dots, \alpha_n], \beta), \beta),$$

and the conditions at level 0 are satisfied. Nevertheless they are contradicted at level 1. To see that, let us take arbitrary elements a and β respectively in $|A|$ and $|!(B \multimap B)|$. Then for any m the following point belongs to R'^* :

$$x_m = ([*], ([2m[(a, a)], m[(a, a)]], \beta), \beta)$$

(using the previous notations, we have fixed $n = 1$ and are now making α_1 vary). For any m we have:

$$x_m^1 = ([*], ([(*, *)], \beta^1), \beta^1),$$

which is independent of m , but:

$$x_m^2 = ([*], [(2m[(a^0, a^0)], m[(a^0, a^0)]], \beta^2, \beta^2)$$

and $s_1(x_m^2) = 2m - m = m$, hence the contradiction.

5 Proof-nets

5.1 Definition

In this section we recall the syntax of proof-nets for LLL introduced in [Gir98] and their normalisation. For background on additive proof-nets (i.e. proof-nets *without* additive boxes) see [Gir96] or the account in [Lau99].

A proof-structure is a labelled graph built from a certain class of nodes that we will give below. For additive proof-nets we need to handle *weights*: elementary weights are boolean variables (denoted as p, q) and weights (denoted as w, w') are products of elementary weights p and negations of elementary weights \bar{p} . If such a product contains a variable p and its negation

we replace it by its value 0. A weight w *depends* on p if p or \bar{p} appears in it. Given a valuation ψ of the variables we denote by $\psi(w)$ (or simply by w if there is no ambiguity) the value of w for this valuation. We write $w \leq w'$ if w is a subproduct of w' .

Now, the nodes and boxes are the following ones:

- node with two conclusions (resp. two premises): ax (resp. cut),
 - nodes with two premises and one conclusion: \otimes , $\&$, \wp ,
 - nodes with one premise and one conclusion: \oplus_1 and \oplus_2 ,
 - node with one conclusion and no premise: \perp , 1 ,
- (for the edges of all these nodes the typing is the natural one)
- *why not* node $?$: an arbitrary number of premises (possibly zero, which is the case of weakening) labelled by the same discharged formula $[A]$ and one conclusion $?A$,
 - additive contraction (denoted as C): at least two premises labelled by the same formula A and one conclusion A ,
 - boxes: see figure 3.

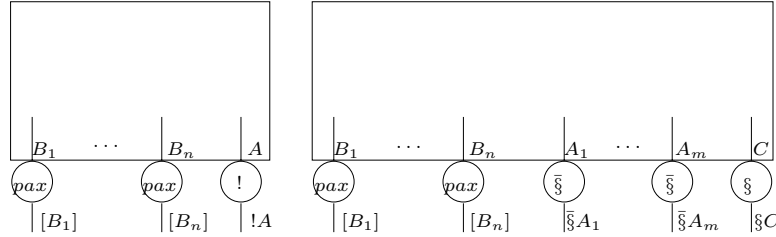


Figure 3: The bang box and paragraph box.

To each $\&$ node of the graph we associate a distinct elementary weight, called its *eigen weight*. To each \perp and $?$ node we associate a non empty set of nodes (not *cut* nodes), called its *jumps*.

A weight $w(A)$ is then associated to each edge A and the following conditions must hold:

- if $A \& B$ is conclusion of a $\&$ node of weight p , then its weight $w = w(A \& B)$ does not depend on p , and the premises of $\&$ have weights $w.p$ and $w.\bar{p}$,

- for an additive contraction node of premises A^1, \dots, A^n and conclusion A : $\sum_{k=1}^n w(A^k) = w(A)$ and for $i \neq j$ $w(A^i) \cdot w(A^j) = 0$,
- for a *why not* node (or \perp node) of conclusion $?A$, if F is one of its premises or the conclusion of one of the jumps, then $w(F) \leq w(?A)$,
- for all other nodes, all adjacent edges must have same weight,
- if w is the weight of the conclusion of a $\&$ node with eigen weight p , and if w' is the weight of an edge depending on p , then $w' \leq w$.
- conditions on boxes: using the notations of figure 3 we should have:
 - for a bang box: $\sum_{i=1}^n w(B_k) = w(A)$,
 - for a paragraph box: there is a partition $I_1 \cup \dots \cup I_k = \{1, \dots, n\}$ such that $\forall j \in \{1, \dots, k\} \sum_{i \in I_j} w(B_i) = w(C)$; furthermore, $\forall i \in \{1, \dots, m\} w(A_i) = w(C)$.

Let $\Gamma = [\Delta]; A_1; \dots; A_n$ be an LLL sequent: each A_i is a block $A_i = A_{i1}, \dots, A_{ik_i}$. An LLL proof-structure of conclusion Γ is a graph satisfying the previous conditions and such that:

$$\sum_{j=1}^{k_i} w(A_{ij}) = 1.$$

A proof-structure coming from the translation of an LLL proof is called a proof-net. Proof-nets can also be characterised by a correctness criterion ([Gir98]).

To illustrate these definitions we give on figure 4 the proof-net corresponding to the proof of the sequent $\vdash ?A^\perp; ?B^\perp; !(A \& B)$, i.e. of the principle $!A \otimes !B \multimap !(A \& B)$ (note that $?A^\perp$ and $?B^\perp$ could also have been given weight 1).

A more “real-scale” example (a proof-net implementing the predecessor) is given in appendix A.

5.2 Normalisation

Let us recall that the normalisation procedure from [Gir98] is a lazy one: we do not reduce commutative additive cuts and only certain exponential cuts (*special* cuts). A *special* cut¹ is an exponential cut such that if one of the

¹This terminology clashes with that used in the thesis of Regnier, where special cut has another meaning.

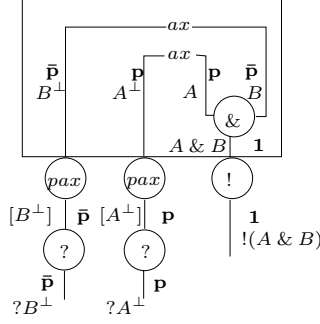


Figure 4: Example of an LLL proof-net.

premises is a $?$ node, then either it is a weakening (it has no premise) or at least one of the premises of this node has weight 1.

- **Contraction:** the elimination of a special contraction cut is carried out in the way described on figure 5 in the case where $w_1 = 1$. In the resulting proof-net R' , the arity of each $?$ node below a discharged formula $[A]$ of $[\Gamma]$ has increased by one: the edge $[A]$ has been replaced by two edges $[A^{(1)}]$ and $[A^{(2)}]$ (this is what we represent by the dashed lines below $[\Gamma^{(1)}]$ and $[\Gamma^{(2)}]$).
- **Weakening:** if $!A$ is cut with a $?$ node with no premise then the box enclosing Θ is simply erased (the $?$ nodes which were conclusion of a $[A]$ in $[\Gamma]$ consequently lose one premise).
- **Paragraph:** on figure 6 we give the paragraph reduction step, which amounts to the merging of two boxes.

We do not recall here the other reduction steps, which are the same as for (multiplicative additive) LL proof-nets ([Gir96]).

5.3 Experiments

For the theory of experiments for LL proof-nets with additive boxes one can see [TdF00] where it is studied in detail. Here we give the modifications needed to handle the LLL proof-nets.

Assume we have chosen an interpretation of atomic formulas. An *experiment* of the proof-net R is an application e sending each edge A to a multiset of visible elements of the web $|A|$ (called its *tag*) and satisfying certain conditions we detail below.

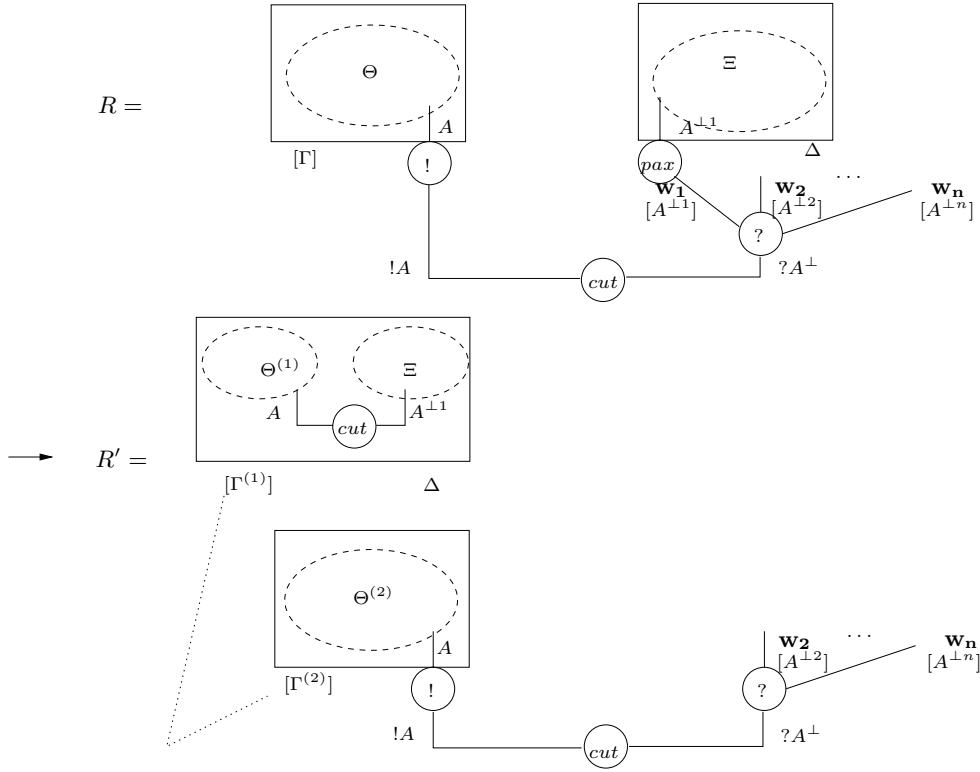


Figure 5: Contraction reduction step.

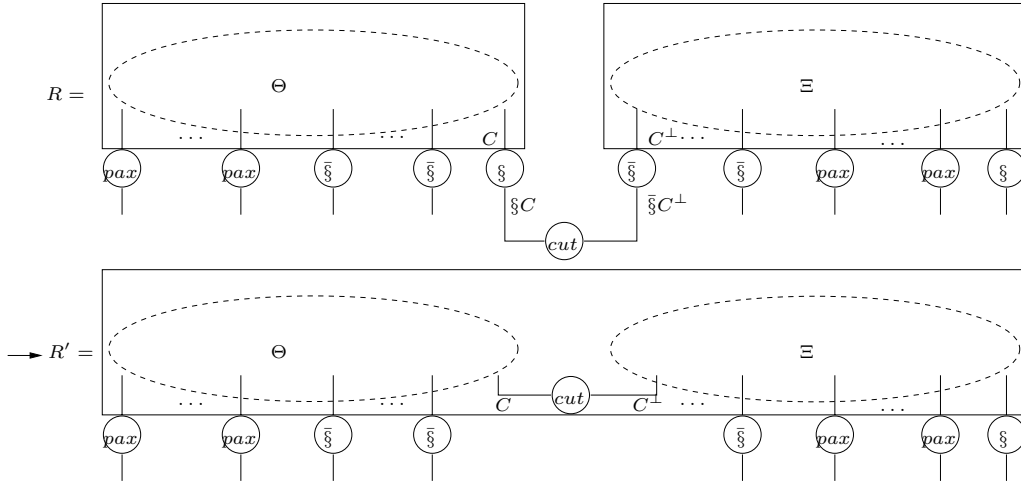


Figure 6: Paragraph reduction step.

We define experiments by induction on the depth of the proof-structure. In the case of a proof-structure of depth 0 (i.e. without any box) we choose a valuation ψ and all edges of weight zero with this valuation are tagged with the empty multiset, the others are tagged with a singleton multiset. Furthermore, the following constraints must hold:

- for an axiom (resp. cut) node of conclusions (resp. premises) A , A^\perp $e(A) = e(A^\perp)$,
- for a \otimes (resp. \wp) node: if $e(A) = [a]$, $e(B) = [b]$ then $e(A \otimes B) = [(a, b)]$ (resp. $e(A \otimes B) = [(a, b)]$),
- for a $\&$ node: $e(A \& B) = e(A) + e(B)$ (note that $e(A) = []$ or $e(B) = []$), and for a \oplus node $e(A \oplus B) = e(A)$ if A is the premise;
- for an additive contraction node: the tag associated to the conclusion is equal to the sum of the tags associated to the premises (at most one of the premises tags is not $[]$),
- for a contraction node whose premises are tagged by $[u_1], \dots, [u_n]$, the conclusion $?A$ is tagged by $[u_1 + \dots + u_n]$; this latter multiset must be a visible element of $|?A|$,
- for an edge labelled by 1 or \perp and of weight non zero, the tag is $[*]$.

Now, for an arbitrary proof-structure R we define an experiment in the following way:

- for each $!$ -box at depth 0 and of content S we take e_1, \dots, e_n (possibly 0) experiments of S and tag each edge A of S with $e(A) = e_1(A) + \dots + e_n(A)$ (empty multiset if $n = 0$); the conclusions of the box are tagged with $e(!C) = [e(C)]$ and $e([B]) = [e(B)]$; $e(C)$ and $e(B)$ should be visible elements respectively of $|!C|$ and $|?B|$;
- for each \S -box at depth 0 at depth 0 and of content S we take e' experiment of S , tag each edge A of S with $e(A) = e'(A)$ and the conclusions of the box with $e([B]) = [e(B)]$, $e(\bar{\S}C) = [e(C)]$, $e(\S D) = [e(D)]$.

To illustrate the notion of experiment we give an example in appendix B.

If e is an experiment of the proof-net R with conclusion $\Gamma = A_1; \dots; A_n$, then it gives a singleton $[a_i]$ for each block A_i . The *result* of the experiment is the element (a_1, \dots, a_n) of $|\Gamma|$.

We define the interpretation R^* of the proof-net R as the set of results of all its experiments. Of course this interpretation coincides with the one we could define directly for sequent calculus proofs: if Π is a sequent calculus proof, R is the proof-net associated and Π^* denotes the interpretation of Π (using the categorical constructions introduced in sections 3.4 and 4.3), then $R^* = \Pi^*$. It follows then from the results of sections 4.2 and 4.3 that:

Proposition 12 *If R is an LLL proof-net, then R^* is a locally bounded stratified clique.*

5.4 Soundness

We are now equipped to prove the soundness of BSCOH for LLL (theorem 11).

Proof : We claim that for any reduction step, if R reduces to R' then $R^* = R'^*$. For multiplicative, additive, axiom and weakening reduction steps there is no novelty compared to the soundness result holding for linear logic and (usual) coherent spaces, so we do not need to go into more detail.

Now, the case of the \S reduction step is simple, so we might as well go on with that. We want to show that if R reduces to R' through this step, then for any experiment of one of these proof-nets, we can give an experiment of the other proof-net with same result. Note that given the modularity of the definition of experiments, we can restrict our attention to the case where R is obtained by a cut between two \S -boxes (the \S rule is the “last rule” of the proof-net) as on figure 6. Take e an experiment of R and define for R' the experiment e' simply by taking for any edge D the tag $e'(D)$ equal to the tag $e(D)$ of the corresponding edge of R . There is no new condition required for e' and so it is indeed a valid experiment of R' , with same result as e . Conversely, any experiment e' of R' yields a corresponding experiment e of R obtained by setting $e(\S C) = [e'(C)]$ and $e(\bar{\S} C^\perp) = [e'(C^\perp)]$ (where $\S C$ and $\bar{\S} C^\perp$ are the premises of the cut).

Now, let us consider the contraction reduction step. We will use the notations of figure 5.

Let e be an experiment of R and γ be the result of e : $\gamma = (u_1, \dots, u_k, \vec{y}, \vec{z})$ where u_1, \dots, u_k are in $[\Gamma]$, \vec{y} is in Δ and \vec{z} are in the other conclusions Δ' of the proof-net.

Then e gives us experiments e_1, \dots, e_n of the box containing Θ with:

$$e_k(A) = e(A_k^\perp).$$

We define an experiment e' of R' in the following way:

- for C edge of $\Theta^{(1)}$: $e'(C) = e_1(C)$,
- for C edge of Ξ : $e'(C) = e(C)$,
- for C edge of $\Theta^{(2)}$: $e'(C) = \sum_{k=2}^n e_k(C)$,
- for other edges C : $e'(C) = e(C)$.

The validity conditions for e' are straightforward, but for the ones dealing with the new contractions on $[\Gamma]$, where there is something to check. By definition of e' we have:

$$e'([\Gamma^{(1)}]) + e'([\Gamma^{(2)}]) = e([\Gamma]),$$

and we know that $e([\Gamma])$ is visible, so the condition for the contraction holds. So do the box conditions.

Conversely, if e' is an experiment of R' with result γ' we define an experiment e of R in the following way:

- for any edge C of Θ : $e(C) = e'(C^{(1)}) + e'(C^{(2)})$,
- for $!A$: $e(!A) = [e'(A^{(1)}) + e'(A^{(2)})]$,
- for $[A^{\perp 1}]$: $e([A^{\perp 1}]) = [e'(A^{\perp 1})]$,
- for other edges C : $e(C) = e'(C)$.

We need to check that $e(!A)$ is visible and that the contraction condition on $e([A^{\perp 1}]), \dots, e([A^{\perp n}])$ holds.

Let \mathcal{A} denote the box enclosing Θ . First we know that $e([\Gamma])$ is visible and as \mathcal{A} is a proof-net we infer that the element of $e(!A)$ is visible. Secondly, by definition of e we have:

$$e(!A) = e(?A^{\perp}) = \left[\sum_{k=1}^n e([A^{\perp k}]) \right].$$

Therefore $\sum_{k=1}^n e([A^{\perp k}])$ is visible and so the condition is satisfied. \square

6 Conclusion and open questions

In this work we showed how the setting of stratified coherent spaces can pinpoint the characteristics of Elementary and Light Linear Logic. It is interesting to note how the multiset interpretation of exponentials is put into use to isolate a quantitative property of certain cliques.

Actually, it should be stressed that the model does keep some intensional features: a (mathematical) function can in fact be realized by several morphisms. This is due both to the typing step which requires *decoration* by modalities ! and §, and to the fact that we are using a multiset interpretation of exponentials, which implies that some information is kept about how many times arguments are used. However we think that this model does provide us with a valuable semantical tool to analyse LLL algorithms which departs significantly from the proper syntactical environment.

A crucial question now is whether this interpretation extends to second-order quantifiers. Indeed these are needed in the syntax to get enough expressivity, i.e. to be able to represent all polytime numerical functions.

Another important question arising is whether the model itself would offer some complexity bound property: is there a notion of computational complexity for stratified cliques and would then the locally stratified cliques fall in a kind of polynomial class? If this is not the case would there be a further constraint on stratified cliques that could ensure such a tractability property?

Going back to the syntax, we saw that our framework incorporates interpretation of ELL and LLL in a unified setting. What about some possible intermediate systems, capturing complexity classes between PTIME and the elementary class? Terui recently introduced a subsystem of Light Affine Logic corresponding to the PSPACE class ([Ter00]), and it would be interesting to find a suitable semantic interpretation of this system.

Finally, from a semantical point of view one can also wonder whether a class of functions corresponds to locally bounded stratified cliques in a similar way stable functions correspond to cliques in the set version of coherent spaces. Thomas Ehrhard and Nuno Bareiro recently characterised the class of functions arising from the multiset coherence model ([BE99]) and maybe their work could be adapted to our stratified setting.

References

- [Abr99] S. Abramsky. Interaction, combinators and complexity. lecture notes, Edinburgh University, 1999.
- [AJM94] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF (extended abstract). In *TACS'94*, number 789 in Lecture Notes in Computer Science. Springer Verlag, 1994.

- [Asp98] Andrea Asperti. Light affine logic. In *"Proceedings of the 13th Symposium on Logic in Computer Science"*. IEEE Computer Society, 1998.
- [BE99] Nuno Barreiro and Thomas Ehrhard. Quantitative semantics revisited. In *Proceedings of the Typed Lambda-Calculi and Applications conference*, volume 1581 of *Lecture Notes in Computer Science*, pages 40–53. Springer Verlag, 1999.
- [BP99] Patrick Baillot and Marco Pedicini. Elementary complexity and the geometry of interaction (Extended Abstract). In *Proceedings of the fourth International Conference on Typed Lambda-Calculi and Applications (TLCA'99)*, volume 1581 of *LNCS*. Springer Verlag, 1999. (full version to appear in *Fundamenta Informaticae*).
- [DJ99] V. Danos and J.-B. Joinet. Linear logic and elementary time. First Workshop on Implicit Computational Complexity (ICC'99), available at <http://www.logique.jussieu.fr/www.danos/>, 1999.
- [Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
- [Gir95] Jean-Yves Girard. Linear logic: its syntax and semantics. In *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1995.
- [Gir96] J.-Y. Girard. Proof-nets: the parallel syntax for proof theory. In *Logic and algebra (Pontignano, 1994)*, volume 180 of *Lecture Notes in Pure and Appl. Math.*, pages 97–124. Dekker, New York, 1996.
- [Gir98] J.-Y. Girard. Light linear logic. *Information and Computation*, 143:175–204, 1998.
- [Hof99] Martin Hofmann. Linear types and non-size-increasing polynomial time computation. In *"Proceedings of the 14th Symposium on Logic in Computer Science"*. IEEE Computer Society, 1999.
- [KOS97] M. I. Kanovich, M. Okada, and A. Scedrov. Phase semantics for light linear logic (extended abstract). In *Mathematical foundations of programming semantics (Pittsburgh, PA, 1997)*, volume 6 of *Electron. Notes Theor. Comput. Sci.* Elsevier, Amsterdam, 1997.
- [Lau99] Olivier Laurent. Polarized proof-nets: proof-nets for LC (extended abstract). In Jean-Yves Girard, editor, *Typed Lambda Calculi and*

Applications '99, volume 1581 of *Lecture Notes in Computer Science*, pages 213–227. Springer, April 1999.

- [MO00] A. S. Murawski and C.H. L. Ong. Discreet games, Light Affine Logic and PTIME computation. to appear in the proceedings of CSL'00, 2000.
- [Rov98] L. Roversi. A Polymorphic Language which is Typable and Polystep. In *Proceedings of the Asian Computing Science Conference (ASIAN'98)*, volume 1538 of *LNCS*, pages 43 – 60. Springer Verlag, December 1998.
- [TdF00] Lorenzo Tortora de Falco. *Réseaux, cohérence et expériences obsessionnelles*. PhD thesis, University Paris 7, january 2000. available at URL <http://www.logique.jussieu.fr/www.tortora/papers.html>.
- [Ter00] K. Terui. Linear logical characterization of polyspace functions. Second Workshop on Implicit Computational Complexity (ICC'00), available at <http://www.loria.fr/~marionjy/ICC00.html>, june 2000.

APPENDICES

A Example of LLL proof-net: the predecessor

We give here the proof-net corresponding to the proof computing the predecessor in [Gir98]. The dashed line stands for a jump. For more readability we use as abbreviation α^\dagger for $\alpha \& \alpha$. The edges with no weight indicated have weight 1.

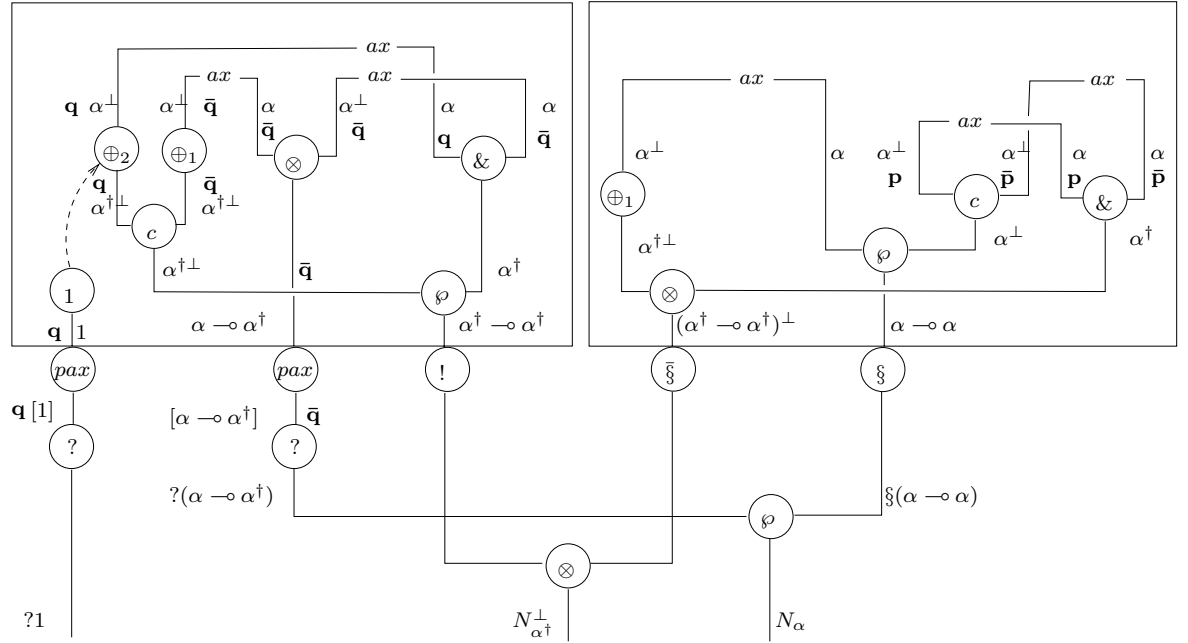


Figure 7: Predecessor.

B Example of experiment

We give in figure 8 an example of experiment for the predecessor proof-net (we omit the formulas and weights to improve the readability). We are considering an interpretation of α by a coherent space with a single point denoted as a (so this is in fact the coherent space 1). We denote the the two points of $|1 \& 1|$ as l and r .

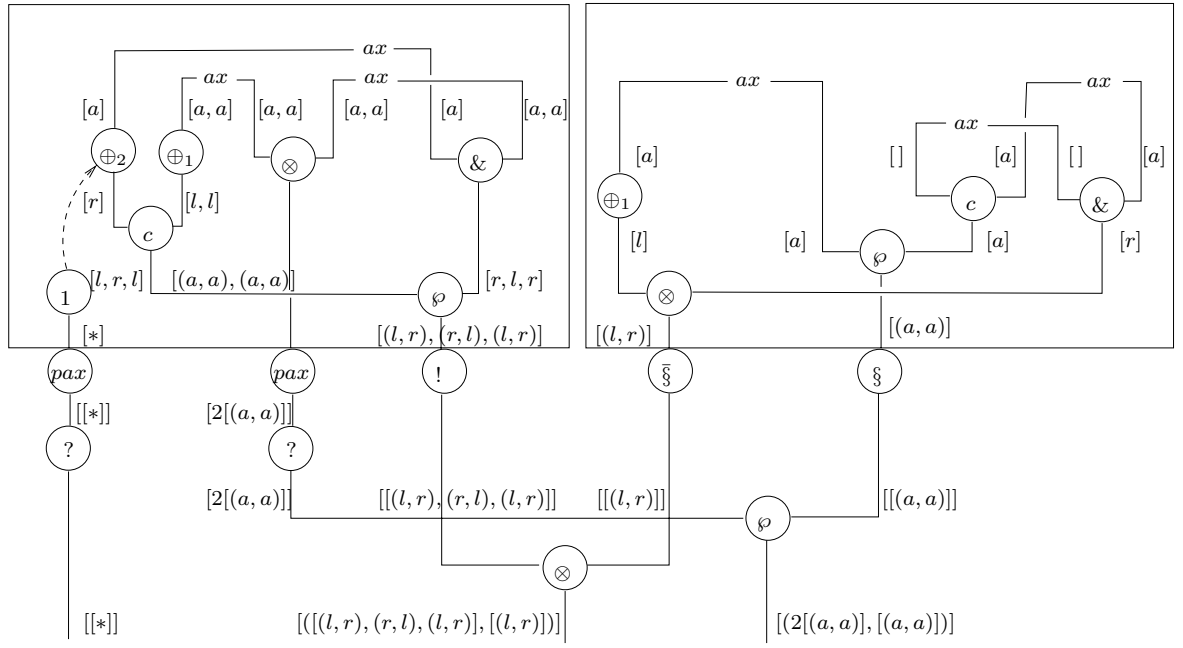


Figure 8: Experiment for the predecessor proof-net.